

Package ‘tmuxr’

May 22, 2020

Title Manage 'tmux' Sessions, Windows, and Panes

Version 0.2.4

Description Create, control, and capture 'tmux' sessions, windows, and panes using a pipeable API.

Depends R (>= 3.3.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL <https://datascienceworkshops.github.io/tmuxr>,
<https://github.com/datascienceworkshops/tmuxr>

BugReports <https://github.com/datascienceworkshops/tmuxr/issues>

SystemRequirements tmux

OS_type unix

Imports processx

Suggests covr, fs, testthat

Collate tmuxr-package.R server.R session.R pane.R window.R send.R
select.R utils.R layout.R inspect.R

RoxygenNote 7.1.0

NeedsCompilation no

Author Jeroen Janssens [aut, cre] (<<https://orcid.org/0000-0002-5878-8924>>),
Data Science Workshops [cph, fnd]

Maintainer Jeroen Janssens <jeroen@datascienceworkshops.com>

Repository CRAN

Date/Publication 2020-05-22 11:40:02 UTC

R topics documented:

attach_pane	3
attach_session	3
attach_window	4
capture_pane	4
clock_mode	5
display_message	6
id	7
index	7
is_active	8
is_installed	8
is_running	9
kill_pane	9
kill_server	9
kill_session	10
kill_window	10
layout	11
list_panes	12
list_sessions	12
list_windows	13
new_pane	13
new_session	14
new_window	15
object_name	16
object_size	17
pipe_pane	18
prop	18
rename_session	19
rename_window	19
resize_pane	20
resize_window	20
select_pane	21
select_window	22
send_keys	23
send_prefix	24
set_option	24
start_server	25
style	25
style_convert	26
swap_pane	26
tmux_command	27
tmux_version	28

attach_pane	<i>Attach to an existing tmux pane</i>
-------------	--

Description

Attach to an existing tmux pane

Usage

```
attach_pane(x, lookup_id = TRUE)
```

Arguments

x	An integer or string indicating the name of an existing pane.
lookup_id	A logical. If FALSE, x is assumed to be a pane id. Default: TRUE.

Value

A tmuxr_pane.

attach_session	<i>Attach to an existing tmux session</i>
----------------	---

Description

Attach to an existing tmux session

Usage

```
attach_session(x, lookup_id = TRUE)
```

Arguments

x	An integer or string indicating the name of an existing window.
lookup_id	A logical. If FALSE, x is assumed to be a session id. Default: TRUE.

Value

A tmuxr_session.

attach_window *Attach to an existing tmux window*

Description

Attach to an existing tmux window

Usage

```
attach_window(x, lookup_id = TRUE)
```

Arguments

x An integer or string indicating the name of an existing window.
lookup_id A logical. If FALSE, x is assumed to be a window id. Default: TRUE.

Value

A tmuxr_window.

capture_pane *Capture the contents of a tmux pane*

Description

Capture the contents of a tmux pane

Usage

```
capture_pane(  
  target = NULL,  
  start = NULL,  
  end = NULL,  
  escape = FALSE,  
  escape_control = FALSE,  
  join = FALSE,  
  cat = FALSE  
)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane. If NULL, the currently active pane is used. Default: NULL.
start, end	An integer or a dash (-). First and last lines to capture. 0 is the first line of the visible pane and negative integers are lines in the history. A dash (-) for start is the start of the history and to end the end of the visible pane. The default is to capture only the visible contents of the pane.
escape	A logical. If TRUE, include escape sequences for text and background attributes. Default: FALSE.
escape_control	A logical. If TRUE, also escape control characters as octal \xxx. Default: FALSE.
join	A logical. If TRUE, join wrapped lines and preserve trailing spaces at each line. Default: FALSE.
cat	A logical. If TRUE, concatenate lines into one string. Default: FALSE.

Value

A vector of strings or one string when cat is TRUE.

See Also

[send_keys\(\)](#)

clock_mode	<i>Display a large clock</i>
------------	------------------------------

Description

Display a large clock

Usage

```
clock_mode(target = NULL)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane. If NULL, the currently active pane is used. Default: NULL.
--------	--

display_message	<i>Display a message</i>
-----------------	--------------------------

Description

Display a message.

Usage

```
display_message(target = NULL, message = NULL, verbose = FALSE, stdout = TRUE)
```

Arguments

target	A session, window, or pane.
message	A string. The message to display. Refer to the FORMATS section of the tmux man page for the format.
verbose	A logical. Print verbose logging as the format is parsed? Default: FALSE.
stdout	A logical. If TRUE, the message is printed to standard output. If FALSE, the message is sent to target. Default: TRUE.

Value

A string if stdout is TRUE, otherwise NULL.

Note

The verbose argument is not supported for tmux version < 2.9 and will be ignored. If verbose is TRUE a warning will be given.

Examples

```
## Not run:
s <- new_session("jazz", height = 12)
display_message(s, "#{window_active}")
display_message(s, "session '#{session_name}' has height #{window_height}.")
kill_session(s)

## End(Not run)
```

id	<i>Identifier of a tmux object</i>
----	------------------------------------

Description

Identifier of a tmux object

Usage

id(target)

Arguments

target A tmuxr_session, tmuxr_window, or tmuxr_pane.

Value

A string.

index	<i>Index of a tmux object</i>
-------	-------------------------------

Description

Index of a tmux object

Usage

index(target)

Arguments

target A tmuxr_window or tmuxr_pane.

Value

An integer.

<code>is_active</code>	<i>Is tmux object active?</i>
------------------------	-------------------------------

Description

Is tmux object active?

Usage

```
is_active(target)
```

Arguments

`target` A `tmuxr_window` or `tmuxr_pane`.

Value

A logical.

<code>is_installed</code>	<i>Is tmux installed correctly?</i>
---------------------------	-------------------------------------

Description

Check whether tmux is installed correctly

Usage

```
is_installed()
```

Value

A logical.

is_running	<i>Is the tmux server running?</i>
------------	------------------------------------

Description

Check whether the tmux server is running.

Usage

```
is_running()
```

Value

A logical.

kill_pane	<i>Kill a tmux pane</i>
-----------	-------------------------

Description

Destroy the given pane. If no panes remain in the containing window, it is also destroyed.

Usage

```
kill_pane(target = NULL, inverse = FALSE)
```

Arguments

target	A tmuxr_pane. If NULL, the currently active pane is used. Default: NULL.
inverse	A logical. If TRUE, kills all but the target pane. Default: FALSE.

kill_server	<i>Kill the tmux server</i>
-------------	-----------------------------

Description

Kill the tmux server

Usage

```
kill_server(silent = TRUE)
```

Arguments

silent	A logical. Should a possible error be suppressed? Default: TRUE.
--------	--

kill_session	<i>Kill a tmux session</i>
--------------	----------------------------

Description

Kill a tmux session

Usage

```
kill_session(target, inverse = FALSE)
```

Arguments

target	A tmuxr_session.
inverse	A logical. If TRUE, kills all but the target session. Default: FALSE.

kill_window	<i>Kill a tmux window</i>
-------------	---------------------------

Description

Kill a tmux window

Usage

```
kill_window(target, inverse = FALSE)
```

Arguments

target	A tmuxr_window.
inverse	A logical. If TRUE, kills all but the target window. Default: FALSE.

layout	<i>Change layout of a tmux window</i>
--------	---------------------------------------

Description

Functions to get and set the layout of a tmux window.

Usage

```
layout(target = NULL)

layout(target) <- value

set_layout(target = NULL, value)

layout_even_horizontal(target = NULL)

layout_even_vertical(target = NULL)

layout_main_horizontal(target = NULL, height = NULL)

layout_main_vertical(target = NULL, width = NULL)

layout_tiled(target = NULL)

layout_next(target = NULL)

layout_previous(target = NULL)

layout_even(target = NULL)

layout_undo(target = NULL)

layout_rotate(target = NULL, reverse = FALSE)
```

Arguments

target	A tmuxr_window. If NULL, use currently active window.
value	A string. Layout definition or name of preset.
width, height	An integer. Width or height of the main pane.
reverse	A logical. If TRUE the direction in which rotate the window is reversed. Default: FALSE.

Value

A string. Layout definition.

list_panes	<i>List tmux panes</i>
------------	------------------------

Description

List tmux panes

Usage

```
list_panes(target = NULL)
```

Arguments

target	A tmuxr_session or tmuxr_window. The parent session or window. If NULL all panes of all sessions are listed. Default: NULL.
--------	---

Value

A list of tmuxr_panes.

list_sessions	<i>List tmux sessions</i>
---------------	---------------------------

Description

List tmux sessions

Usage

```
list_sessions()
```

Value

A list of tmuxr_sessions.

list_windows	<i>List tmux windows</i>
--------------	--------------------------

Description

List tmux windows

Usage

```
list_windows(target = NULL)
```

Arguments

target	Name of parent tmuxr_session. If NULL (default), all windows of all sessions are listed.
--------	--

Value

A list of tmuxr_windows.

new_pane	<i>Create a new tmux pane</i>
----------	-------------------------------

Description

Create a new pane by splitting target.

Usage

```
new_pane(
  target = NULL,
  vertical = TRUE,
  size = NULL,
  before = FALSE,
  full = FALSE,
  start_directory = NULL,
  shell_command = NULL
)
```

```
split_window(
  target = NULL,
  vertical = TRUE,
  size = NULL,
  before = FALSE,
  full = FALSE,
  start_directory = NULL,
  shell_command = NULL
)
```

Arguments

target	A tmux_session, tmuxr_window, or tmuxr_pane.
vertical	A logical. If TRUE split target vertically, otherwise horizontally. Default: TRUE.
size	A numeric. The size of the new pane in lines (for vertical splits) or characters (for horizontal splits). If less than 0, size is interpreted as a percentage. Default: NULL.
before	A logical. If TRUE, create the new pane to the left of or above the active pane. Default: FALSE.
full	A logical. If TRUE, create a new pane spanning the full window width (for vertical splits) or full window height (for horizontal splits), instead of splitting the active pane. Default: FALSE.
start_directory	A string. Working directory this pane is run in.
shell_command	A string. Shell command to be invoked when creating the pane. If NULL, the default shell is used.

Value

A tmuxr_pane.

new_session	<i>Create a new tmux session</i>
-------------	----------------------------------

Description

Create a new tmux session with name name.

Usage

```
new_session(
  name = NULL,
  window_name = NULL,
  start_directory = NULL,
  width = 80,
  height = 24,
  detached = TRUE,
  shell_command = NULL
)
```

Arguments

name	A string. Name of the session. If NULL, the name determined by tmux, which is the next unused integer (by default starting at 0). Default: NULL.
window_name	A string. Name of initial window.

`start_directory` A string. Working directory this session is run in.

`width` An integer. Width of initial window. Default: 80.

`height` An integer. Height of initial window. Default: 24.

`detached` A logical. If FALSE, the R interpreter waits for the session to be killed. Default: TRUE.

`shell_command` A string. Shell command to be invoked when creating the session.

Value

A `tmuxr_session`.

Note

Prior to `tmux` version 2.6, the actual height is one line less than `height`.

Examples

```
## Not run:
s <- new_session("foo", shell_command = "bash", height = 10)
list_sessions()
kill_session(s)

## End(Not run)
```

<code>new_window</code>	<i>Create a new tmux window</i>
-------------------------	---------------------------------

Description

Create a new `tmux` window with name `name`. When the shell command completes, the window closes.

Usage

```
new_window(
  target = NULL,
  name = NULL,
  focus = TRUE,
  start_directory = NULL,
  shell_command = NULL
)
```

Arguments

target	A tmuxr_session or tmuxr_window. If NULL, the last session is used.
name	A string. Name of the window. If NULL, the name determined by the shell command. Default: NULL.
focus	A logical. If FALSE, the session does not make the new window the current window. Default: TRUE.
start_directory	A string. Working directory this window is run in.
shell_command	A string. Shell command to be invoked when creating the window. If NULL, the default shell is used.

Value

A tmuxr_window.

Examples

```
## Not run:
s <- new_session()
new_window()
kill_session(s)

## End(Not run)
```

object_name	<i>Name of a tmux object</i>
-------------	------------------------------

Description

Functions to get and set the name of a tmux session, window, and pane.

Usage

```
name(target)

name(target) <- value

set_name(target, value)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane.
value	A string. New name.

Value

A string.

See Also

[rename_session\(\)](#), [rename_window\(\)](#)

object_size	<i>Width and height of a tmux object</i>
-------------	--

Description

Functions to get and set the width and height of a tmux session, window, and pane.

Usage

```
width(target)
```

```
width(target) <- value
```

```
set_width(target, value)
```

```
height(target)
```

```
height(target) <- value
```

```
set_height(target, value)
```

Arguments

target A tmuxr_session, tmuxr_window, or tmuxr_pane.

value An integer.

Value

An integer. The new width or height.

Note

The size of a tmuxr_pane can only be changed if there are other panes in the same window.

See Also

[resize_window\(\)](#), [resize_pane\(\)](#)

pipe_pane	<i>Pipe contents of a tmux pane to a shell command</i>
-----------	--

Description

Pipe contents of a tmux pane to a shell command

Usage

```
pipe_pane(
    target = NULL,
    shell_command = NULL,
    stdout = TRUE,
    stdin = FALSE,
    open = FALSE
)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane.
shell_command	String. If NULL, the current pipe (if any) is closed. Default: NULL.
stdout	Logical. Connect standard output of pane to shell_command? Default: TRUE.
stdin	Logical. Connect standard input of pane to shell_command? Default: FALSE.
open	Logical. Only open a new pipe if no previous pipe exists. Default: FALSE.

prop	<i>Properties of a tmux object</i>
------	------------------------------------

Description

Get the properties of a tmux object, including sessions, windows, and panes.

Usage

```
prop(target = NULL, property)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_session. Default: NULL.
property	A string.

Value

A string.

Examples

```
## Not run:  
s <- new_session()  
prop(s, "session_created")  
kill_session(s)  
  
## End(Not run)
```

rename_session	<i>Rename a tmux session</i>
----------------	------------------------------

Description

Rename a tmux session

Usage

```
rename_session(target, value)
```

Arguments

target	A tmuxr_session.
value	String indicating the new name of the session.

Value

A tmuxr_session.

rename_window	<i>Rename a tmux window</i>
---------------	-----------------------------

Description

Rename a tmux window

Usage

```
rename_window(target, value)
```

Arguments

target	A tmuxr_window.
value	String indicating the new name of the window.

resize_pane	<i>Resize a tmux pane</i>
-------------	---------------------------

Description

Resize a tmux pane

Usage

```
resize_pane(target, width = NULL, height = NULL)
```

Arguments

target	A tmuxr_pane.
width	An integer. Default: NULL.
height	An integer. Default: NULL.

resize_window	<i>Resize a tmux window</i>
---------------	-----------------------------

Description

Resize a tmux window

Usage

```
resize_window(target, width = NULL, height = NULL)
```

Arguments

target	A tmuxr_session or tmuxr_window.
width	An integer. Default: NULL.
height	An integer. Default: NULL.

select_pane	<i>Select a tmux pane</i>
-------------	---------------------------

Description

Note that selecting a pane from a non-active window does not select that window.

Usage

```
select_pane(target = NULL, token = NULL)
select_pane_active(target = NULL)
select_pane_last(target = NULL)
select_pane_next(target = NULL)
select_pane_previous(target = NULL)
select_pane_top(target = NULL)
select_pane_bottom(target = NULL)
select_pane_left(target = NULL)
select_pane_right(target = NULL)
select_pane_top_left(target = NULL)
select_pane_top_right(target = NULL)
select_pane_bottom_left(target = NULL)
select_pane_bottom_right(target = NULL)
select_pane_up_of(target = NULL)
select_pane_down_of(target = NULL)
select_pane_left_of(target = NULL)
select_pane_right_of(target = NULL)
```

Arguments

target	A <code>tmuxr_session</code> , <code>tmuxr_window</code> , or <code>tmuxr_pane</code> . If <code>NULL</code> , the currently active pane is used. Default: <code>NULL</code> .
--------	--

token A string. Default NULL.

Value

A tmuxr_pane.

select_window	<i>Select a tmux window</i>
---------------	-----------------------------

Description

Select a tmux window

Usage

```
select_window(target = NULL, token = NULL)
```

```
select_window_active(target = NULL)
```

```
select_window_last(target = NULL)
```

```
select_window_next(target = NULL)
```

```
select_window_previous(target = NULL)
```

```
select_window_start(target = NULL)
```

```
select_window_end(target = NULL)
```

Arguments

target A tmuxr_session or tmuxr_window. If NULL, the last session is used. Default: NULL.

token A string. Default NULL.

Value

A tmuxr_window.

send_keys	<i>Send keys to a tmux pane</i>
-----------	---------------------------------

Description

Send keys to a tmux pane

Usage

```
send_keys(target = NULL, ..., literal = FALSE, count = 1L)
```

Arguments

target	A <code>tmuxr_session</code> , <code>tmuxr_window</code> , or <code>tmuxr_pane</code> . If <code>NULL</code> , the currently active pane is used. Default: <code>NULL</code> .
...	Strings. Keys to send.
literal	A logical. If <code>TRUE</code> , key name lookup is disabled and the keys are processed as literal UTF-8 characters. Default: <code>FALSE</code> .
count	An integer. Number of times the keys are sent. Default: <code>1L</code> .

See Also

[capture_pane\(\)](#)

Examples

```
## Not run:  
s <- new_session(shell_command = "cat")  
send_keys(s, "Speak", "Space", "friend")  
send_keys(s, "BSpace", count = 6L)  
send_keys(s, "mellon and ")  
send_keys(s, "enter", "!", literal = TRUE)  
send_keys(s, "enter", literal = FALSE)  
capture_pane(s, start = 0L, end = 1L)  
kill_session(s)  
  
## End(Not run)
```

send_prefix	<i>Send prefix to a tmux pane</i>
-------------	-----------------------------------

Description

Send prefix to a tmux pane

Usage

```
send_prefix(target = NULL, secondary = FALSE)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane. If NULL, the currently active pane is used. Default: NULL.
secondary	A logical. If TRUE, send secondary prefix. Default: FALSE.

set_option	<i>Set option</i>
------------	-------------------

Description

Set option

Usage

```
set_option(
  target = NULL,
  option,
  value = NULL,
  type = c("session", "window", "pane", "server"),
  append = FALSE,
  expand = FALSE,
  global = FALSE,
  unset = FALSE,
  override = TRUE
)
```

Arguments

target	A tmuxr_session, tmuxr_window, or tmuxr_pane. Default: NULL.
option	A string.
value	A string. Default: NULL.
type	A string. Type of option. Default: session.

append	A logical. Default: FALSE.
expand	A logical. Default: FALSE.
global	A logical. Default: FALSE.
unset	A logical. Default: FALSE.
override	A logical. Default: TRUE.

start_server	<i>Start the tmux server</i>
--------------	------------------------------

Description

Start the tmux server

Usage

```
start_server()
```

style	<i>Style of a tmux window or pane</i>
-------	---------------------------------------

Description

Functions to get and set the layout of a tmux window or pane.

Usage

```
style(target)
```

```
style(target) <- value
```

```
set_style(target, value)
```

Arguments

target A tmuxr_window or tmuxr_pane.

value A named list. Style definition.

style_convert	<i>Format and parse style strings</i>
---------------	---------------------------------------

Description

Functions to convert a style string to a named list and back.

Usage

```
strfstyle(...)
```

```
strpstyle(x)
```

Arguments

...	Named strings and logicals or one named list. Colors and attributes.
x	A string.

Value

A string or named list.

Examples

```
## Not run:
strfstyle(fg = "red", bg = "#00ff00", blink = TRUE, align = FALSE)
strpstyle("fg=red,bg=#00ff00,blink,noalign")

## End(Not run)
```

swap_pane	<i>Swap two tmux windows or panes</i>
-----------	---------------------------------------

Description

Swap two tmux windows or panes

Usage

```
swap_pane(from = NULL, to = NULL, reverse = FALSE, select = TRUE)
```

```
swap_window(from = NULL, to = NULL, select = TRUE)
```

Arguments

from	A tmuxr_window or tmuxr_pane.
to	A tmuxr_window or tmuxr_pane.
reverse	A logical.
select	A logical. If FALSE, do not select the new window or pane. Default: TRUE.

tmux_command	<i>Execute a tmux command</i>
--------------	-------------------------------

Description

Execute a tmux command

Usage

```
tmux_command(command, ..., .silent = FALSE)
```

Arguments

command	A string.
...	Strings. Command flags.
.silent	A Logical. Default: FALSE.

Value

A vector of strings.

Examples

```
## Not run:  
s <- new_session()  
split_window(s)  
tmux_command("list-panes", "-t" , "0")  
kill_session(s)  
  
## End(Not run)
```

tmux_version	<i>Get tmux version</i>
--------------	-------------------------

Description

Get the version of tmux.

Usage

```
tmux_version(as_numeric = TRUE)
```

Arguments

`as_numeric` A logical. If TRUE, the version number is returned as a numeric. Default: TRUE.

Value

A numeric or string.

Examples

```
## Not run:  
tmux_version()  
tmux_version() >= 2.8  
tmux_version(as_numeric = FALSE)  
  
## End(Not run)
```

Index

attach_pane, 3
attach_session, 3
attach_window, 4

capture_pane, 4
capture_pane(), 23
clock_mode, 5

display_message, 6

height (object_size), 17
height<- (object_size), 17

id, 7
index, 7
is_active, 8
is_installed, 8
is_running, 9

kill_pane, 9
kill_server, 9
kill_session, 10
kill_window, 10

layout, 11
layout<- (layout), 11
layout_even (layout), 11
layout_even_horizontal (layout), 11
layout_even_vertical (layout), 11
layout_main_horizontal (layout), 11
layout_main_vertical (layout), 11
layout_next (layout), 11
layout_previous (layout), 11
layout_rotate (layout), 11
layout_tiled (layout), 11
layout_undo (layout), 11
list_panes, 12
list_sessions, 12
list_windows, 13

name (object_name), 16

name<- (object_name), 16
new_pane, 13
new_session, 14
new_window, 15

object_name, 16
object_size, 17

pipe_pane, 18
prop, 18

rename_session, 19
rename_session(), 17
rename_window, 19
rename_window(), 17
resize_pane, 20
resize_pane(), 17
resize_window, 20
resize_window(), 17

select_pane, 21
select_pane_active (select_pane), 21
select_pane_bottom (select_pane), 21
select_pane_bottom_left (select_pane),
21
select_pane_bottom_right (select_pane),
21
select_pane_down_of (select_pane), 21
select_pane_last (select_pane), 21
select_pane_left (select_pane), 21
select_pane_left_of (select_pane), 21
select_pane_next (select_pane), 21
select_pane_previous (select_pane), 21
select_pane_right (select_pane), 21
select_pane_right_of (select_pane), 21
select_pane_top (select_pane), 21
select_pane_top_left (select_pane), 21
select_pane_top_right (select_pane), 21
select_pane_up_of (select_pane), 21
select_window, 22

`select_window_active` (`select_window`), 22
`select_window_end` (`select_window`), 22
`select_window_last` (`select_window`), 22
`select_window_next` (`select_window`), 22
`select_window_previous` (`select_window`),
22
`select_window_start` (`select_window`), 22
`send_keys`, 23
`send_keys()`, 5
`send_prefix`, 24
`set_height` (`object_size`), 17
`set_layout` (`layout`), 11
`set_name` (`object_name`), 16
`set_option`, 24
`set_style` (`style`), 25
`set_width` (`object_size`), 17
`split_window` (`new_pane`), 13
`start_server`, 25
`strfstyle` (`style_convert`), 26
`strpstyle` (`style_convert`), 26
`style`, 25
`style<-` (`style`), 25
`style_convert`, 26
`swap_pane`, 26
`swap_window` (`swap_pane`), 26

`tmux_command`, 27
`tmux_version`, 28

`width` (`object_size`), 17
`width<-` (`object_size`), 17